

Lógica de programação

Profº Jânio Eduardo

janio.vasconcellos@gmail.com

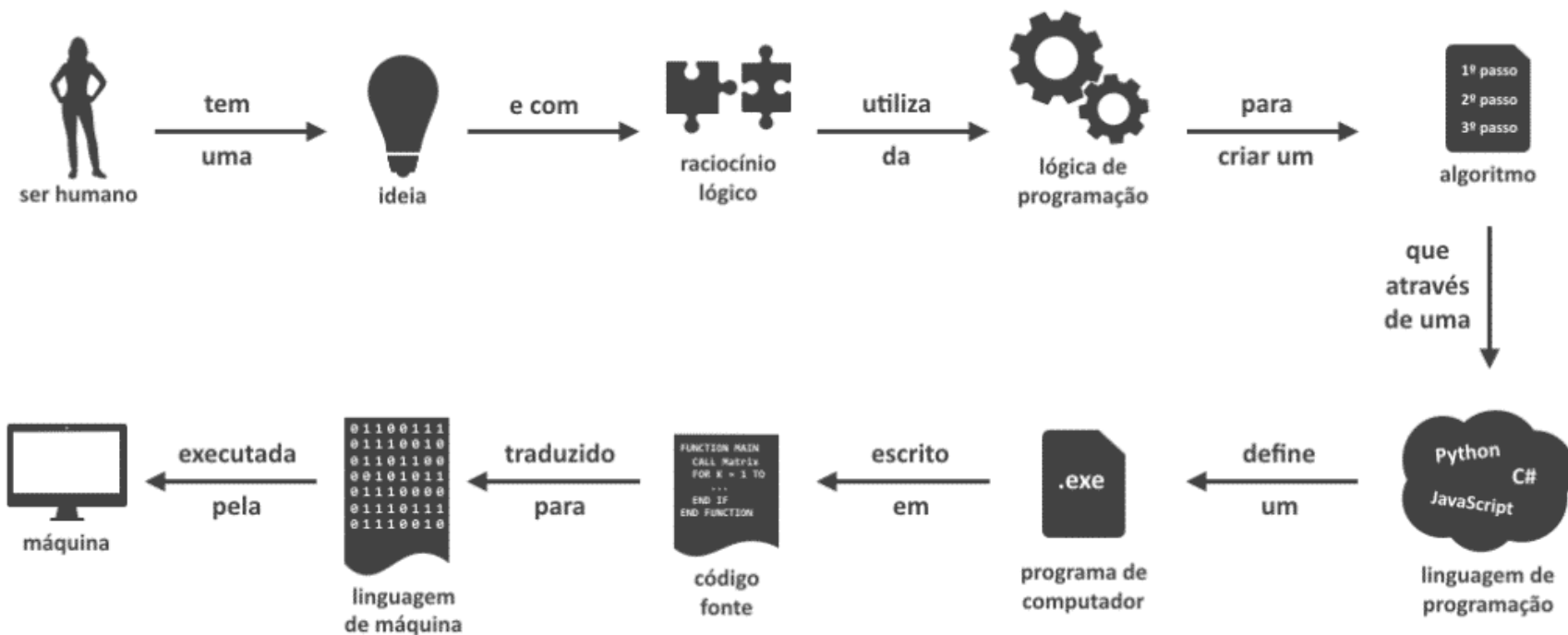
- ▶ Objetivo geral
- ▶ Introdução Lógica de programação
 - ▶ Conceitos básicos de programação
 - ▶ Evolução das linguagens de programação
 - ▶ Ambiente de desenvolvimento e ferramentas de programação
- ▶ Resumo da aula

Objetivo geral

Entender qual contribuição será aplicada
com a aplicação da lógica de programação e
confeção de algoritmos

Conheça como é uma
fábrica de software

Fluxo de trabalho



Fábrica de Software

Processos

1 Planejamento



2 Design



3 Desenvolvimento



4 Testes



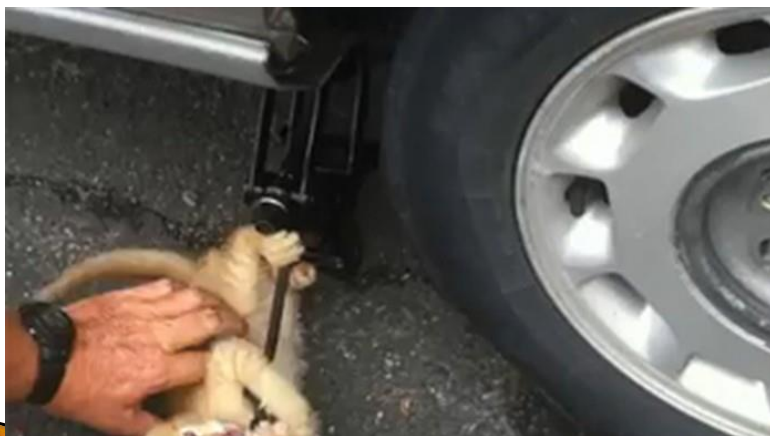
6 Suporte



5 Implantação



Problemas do cotidiano para o pensamento lógico





Vamos com calma



Conceitos básicos

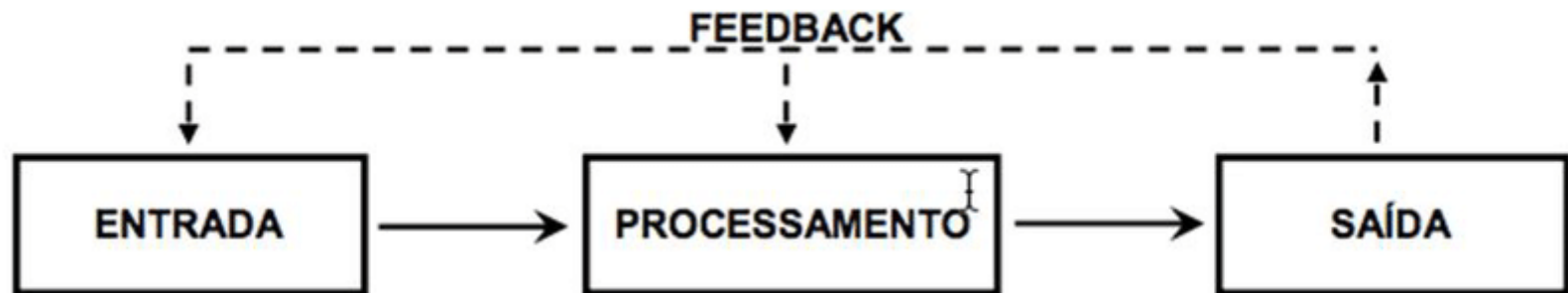
- ▶ Dados;
- ▶ Processamento;
- ▶ Algoritmos;
- ▶ Linguagens de programação;
- ▶ Compilação e Interpretação;
- ▶ Estrutura de dados;
- ▶ Programação Orientada a objetos;
- ▶ Testes e depuração.

Dados

- ▶ Informações que podem ser armazenadas e manipuladas por um computador, como texto, números, imagens e sons

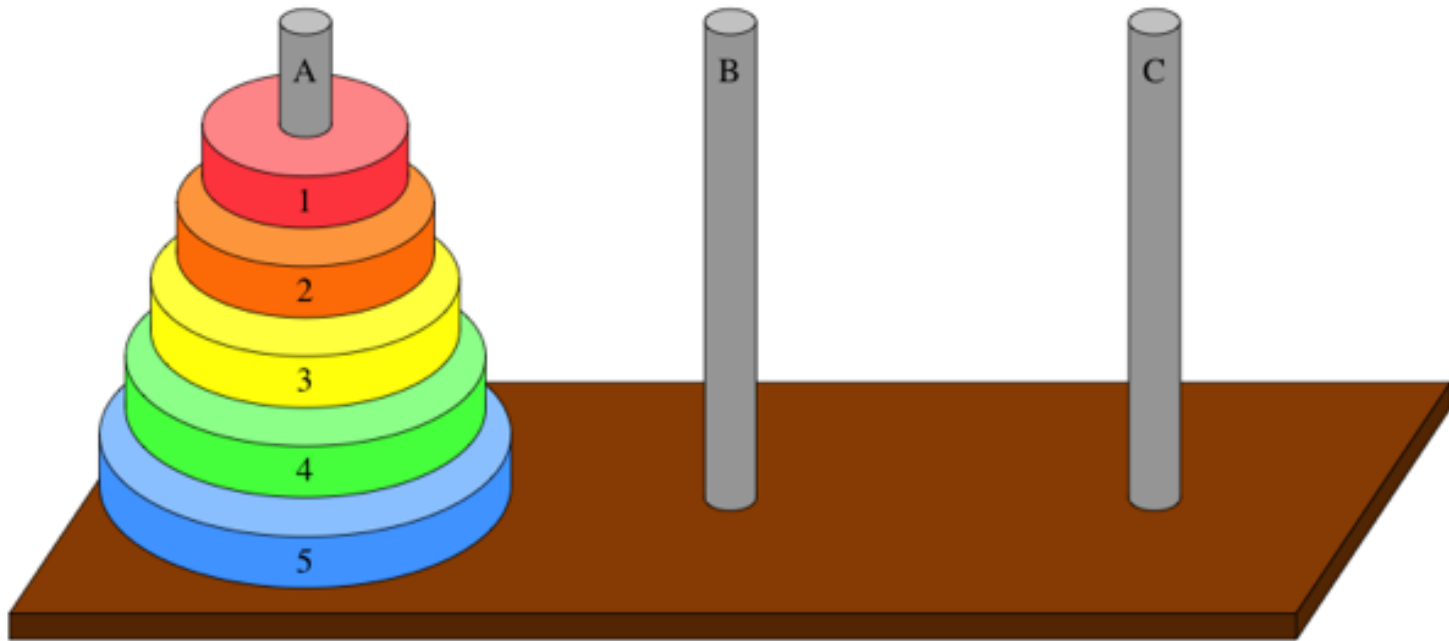
Processamento

- ▶ A capacidade do computador de executar tarefas e operações em dados, com base em instruções específicas.



Algoritmos

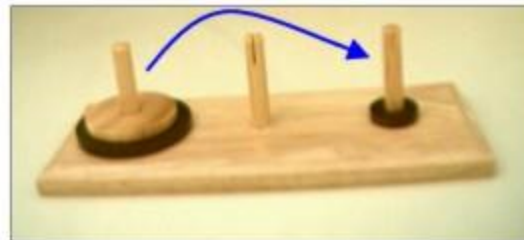
- ▶ Uma sequência de instruções que descrevem uma solução para um problema ou tarefa.
- ▶ Ex: Estudo de caso das Torres de Hanoi



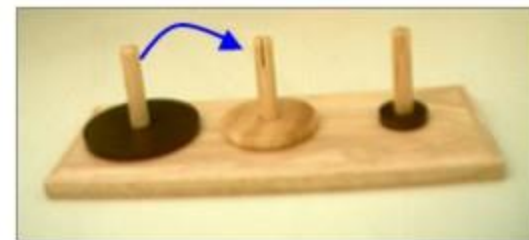
Torres de hanoi pseudo fluxogramas



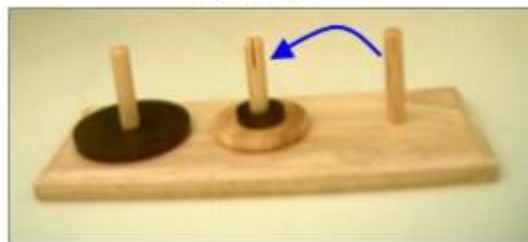
início



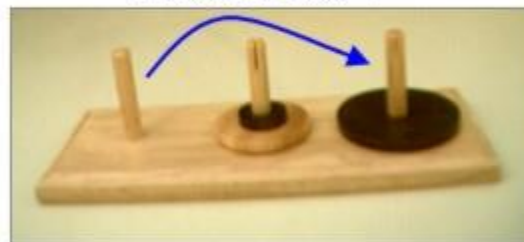
movimento 1



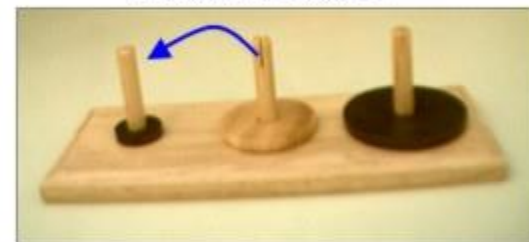
movimento 2



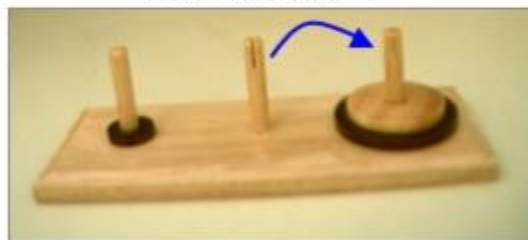
movimento 3



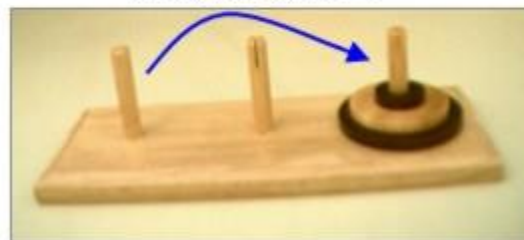
movimento 4



movimento 5



movimento 6



movimento 7

Fig. 4 – Transferência da torre real do pino da esquerda para o pino da direita.

Explicando em Fluxogramas

Algoritmo

1. Mover um anel da haste A para haste B;
2. Mover um anel da haste A para haste C;
3. Mover um anel da haste B para haste C;
4. Mover um anel da haste A para haste B;
5. Mover um anel da haste C para haste A;
6. Mover um anel da haste C para haste B;
7. Mover um anel da haste A para haste B;

Explicação

- ▶ O algoritmo define uma sequencia de instruções, há outras formas de resolver;
- ▶ A depender do programa pode ter maior ou menor uso do processamento;

Linguagens de programação

Histórico e definições

Linguagens de programação

- 1951 - Regional Assembly Language
- 1952 - Autocode
- 1954 - FORTRAN
- 1955 - FLOW-MATIC (antecessor do COBOL)
- 1957 - COMTRAN (antecessor do COBOL)
- 1958 - LISP
- 1958 - ALGOL 58
- 1959 - FACT (antecessor do COBOL)
- 1959 - COBOL
- 1962 - APL
- 1962 - Simula
- 1964 - BASIC
- 1964 - PL/I
- 1970 - Pascal
- 1970 - Forth
- 1972 - C
- 1972 - Smalltalk
- 1972 - Prolog
- 1973 - ML
- 1978 - SQL
- 1983 - Ada
- 1983 - C++
- 1984 - MATLAB
- 1985 - Eiffel
- 1986 - Objective-C
- 1987 - Perl
- 1989 - FL (Backus)
- 1990 - Haskell
- 1991 - Python
- 1991 - Java
- 1993 - R
- 1993 - Lua
- 1994 - ANSI Common Lisp
- 1995 - Delphi (Object Pascal)
- 1995 - JavaScript
- 1995 - PHP
- 1995 - Ruby
- 2000 - C#
- 2000 - ActionScript
- 2002 - Scratch
- 2008 - JavaFX Script
- 2006 - PowerShell
- 2009 - Go
- 2012 - Julia
- 2012 - TypeScript
- 2014 - Swift
- 2014 - Hack
- 2017 - Q#

Linguagem COBOL

- ▶ COBOL (COmmon Business Oriented Language) é uma linguagem de programação de alto nível que foi desenvolvida originalmente na década de 1950 por um comitê de programadores de várias empresas de computadores, liderados por Grace Hopper. A linguagem COBOL foi projetada especificamente **para processar dados comerciais e foi amplamente utilizada em sistemas de processamento de transações, folha de pagamento e contabilidade durante as décadas de 1960 e 1970**. A linguagem COBOL é conhecida por ser altamente legível e ter uma sintaxe simples que facilita a manutenção e o suporte de grandes sistemas empresariais. Ela também foi projetada para ser altamente portátil, permitindo que os programas escritos em COBOL fossem executados em diferentes sistemas operacionais e plataformas de hardware. Apesar de ter sido criada há muitos anos, a linguagem COBOL ainda é amplamente utilizada em muitas organizações empresariais em todo o mundo, especialmente em sistemas legados. Entretanto, **muitos programadores COBOL estão se aposentando**, e há uma **crescente necessidade de novos programadores** aprenderem a linguagem para manter e modernizar esses sistemas legados.



Linguagem C

- ▶ A linguagem C é uma linguagem de programação de alto nível criada na década de 1970 por Dennis Ritchie para desenvolver o sistema operacional Unix. Ela é uma linguagem de propósito geral e pode ser usada para desenvolver uma ampla variedade de aplicativos, **desde aplicativos de sistema até jogos, aplicativos para dispositivos móveis e aplicações web.** A linguagem C é conhecida por ser rápida, eficiente e poderosa, oferecendo recursos como ponteiros, gerenciamento de memória, suporte a bibliotecas padrão e uma sintaxe simples e intuitiva. É frequentemente usada em sistemas operacionais, compiladores, bancos de dados e outras aplicações de baixo nível. A linguagem C também é considerada a base para muitas outras linguagens de programação modernas, como C++, C# e Java.



Linguagem SQL

- ▶ SQL (Structured Query Language) é uma linguagem de programação utilizada para gerenciar dados em sistemas de bancos de dados relacionais. Ela é usada para criar, modificar e consultar bancos de dados, bem como para definir e modificar as estruturas de tabelas e outros objetos relacionais dentro do banco de dados. A linguagem SQL é composta por um conjunto de comandos que permitem que os usuários executem operações de CRUD (criação, leitura, atualização e exclusão) em bancos de dados, como inserir, atualizar ou recuperar informações de tabelas, definir restrições de integridade, criar índices e tabelas, entre outras tarefas. A linguagem SQL é usada por muitos sistemas de gerenciamento de bancos de dados, como MySQL, Oracle, Microsoft SQL Server, PostgreSQL e SQLite.



Linguagem C++

- ▶ C++ é uma linguagem de programação orientada a objetos que foi criada como uma extensão da linguagem C original. Ela foi desenvolvida na década de 1980 por Bjarne Stroustrup como uma forma de adicionar recursos de orientação a objetos e outras funcionalidades à linguagem C. A linguagem C++ mantém muitos dos recursos da linguagem C, como sua sintaxe simples e eficiente, mas adiciona suporte a recursos de programação orientada a objetos, como classes, objetos, herança, polimorfismo e encapsulamento. Ela é usada em muitos tipos de aplicativos, incluindo jogos, sistemas de controle, software de simulação, aplicativos de desktop e aplicativos web. A linguagem C++ é frequentemente usada em sistemas operacionais, drivers de dispositivos, sistemas embarcados e outras aplicações de baixo nível. Ela é conhecida por ser uma linguagem de programação de alto desempenho e é amplamente utilizada na indústria de software.



Linguagem Objective-C

- ▶ Objective-C é uma linguagem de programação orientada a objetos que foi criada na década de 1980 por Brad Cox e Tom Love. Ela é uma extensão da linguagem de programação C, que adiciona suporte a recursos de programação orientada a objetos, como classes, objetos, herança, polimorfismo e encapsulamento. Objective-C foi originalmente desenvolvida para ser usada em sistemas operacionais Unix e é mais conhecida por ser a principal linguagem de programação usada para desenvolver aplicativos para o sistema **operacional macOS e para dispositivos iOS, como iPhones e iPads**. Objective-C é frequentemente usada em conjunto com o framework Cocoa, que fornece uma estrutura de desenvolvimento de aplicativos para sistemas macOS e iOS. A linguagem Objective-C foi amplamente utilizada no passado, mas em 2014, a Apple anunciou que a linguagem Swift seria a linguagem principal para o desenvolvimento de aplicativos iOS e macOS. No entanto, **muitos aplicativos legados ainda são escritos em Objective-C** e a linguagem continua sendo relevante em algumas áreas de desenvolvimento de software.



Swift

- ▶ Swift é uma linguagem de programação de alto nível criada pela Apple em 2014 para o desenvolvimento de aplicativos para sistemas **macOS, iOS, watchOS e tvOS**. Ela foi projetada para ser uma linguagem moderna, rápida, segura e fácil de usar, com recursos como inferência de tipo de dados, closures, tuplas e opcionalidade. A linguagem Swift é usada para criar aplicativos em uma ampla variedade de áreas, desde jogos e aplicativos de mídia até aplicativos empresariais e educacionais. Ela é compatível com o ambiente de desenvolvimento integrado (IDE) Xcode e oferece uma ampla variedade de ferramentas de desenvolvimento, incluindo depuração avançada, testes de unidade, análise de código e perfis de desempenho. A linguagem Swift é **conhecida por ser fácil de aprender**, especialmente para desenvolvedores que já estão familiarizados com outras linguagens de programação, como C++, Java ou Python. Desde o seu lançamento, a linguagem Swift tem crescido rapidamente em popularidade e se tornou uma das **linguagens de programação mais usadas no desenvolvimento de aplicativos móveis**.

Linguagem Python

- ▶ Python é uma linguagem de programação de alto nível, interpretada e orientada a objetos que foi criada por Guido van Rossum em 1991. Ela é conhecida por ter uma **sintaxe clara e concisa que facilita a leitura e a escrita de código**. Python é uma linguagem de uso geral e pode ser usada em muitas áreas, como desenvolvimento de software, análise de dados, aprendizado de máquina, ciência de dados, automação de tarefas e scripting. Ela é popular entre programadores iniciantes, bem como programadores experientes, devido à sua facilidade de uso e versatilidade. Python tem uma grande variedade de bibliotecas e frameworks disponíveis que permitem desenvolver aplicativos complexos rapidamente, **tais como Flask, Django, Pandas, Numpy, Matplotlib e TensorFlow**. Além disso, a comunidade de programadores Python é muito ativa e oferece suporte por meio de fóruns de discussão, sites de perguntas e respostas, bem como tutoriais e documentação online. **Python é uma das linguagens de programação mais populares do mundo e continua a crescer em popularidade a cada ano.**

Linguagem JAVA

- ▶ Java é uma linguagem de programação orientada a objetos de alto nível que foi criada pela empresa americana Sun Microsystems em 1995. Ela foi projetada para ser portátil, segura e fácil de usar, e rapidamente se tornou popular entre programadores em todo o mundo. Java é frequentemente **usado para desenvolver aplicativos de software para uma ampla variedade de plataformas, incluindo computadores desktop, servidores web, dispositivos móveis e até mesmo dispositivos de Internet das Coisas (IoT).** Além disso, a linguagem Java é conhecida por sua grande quantidade de bibliotecas e frameworks, como o Spring Framework e o Hibernate, que **ajudam os programadores a desenvolver aplicativos complexos rapidamente e com facilidade.** Java é executado em uma máquina virtual Java (JVM), que é responsável por interpretar e executar o código Java em diferentes sistemas operacionais e plataformas. A linguagem Java é amplamente **utilizada em muitas áreas, como desenvolvimento de jogos, desenvolvimento web, automação de tarefas e desenvolvimento de aplicativos móveis.** Ela também é uma das linguagens mais populares entre desenvolvedores corporativos devido à sua segurança, confiabilidade e escalabilidade.



Linguagem Javascript

- ▶ JavaScript é uma linguagem de programação de alto nível que foi criada em 1995 pela Netscape Communications Corporation. Ela foi originalmente **projetada para ser usada em páginas web** para torná-las mais interativas e dinâmicas, permitindo que os desenvolvedores criem efeitos visuais, animações, formulários dinâmicos e outras **funcionalidades avançadas do lado do cliente**. No entanto, com o tempo, o JavaScript evoluiu para ser uma linguagem de programação completa e pode ser usado para desenvolver aplicativos de software em **várias plataformas, incluindo web, desktop e dispositivos móveis**. O JavaScript é interpretado pelo navegador web e não precisa ser compilado, tornando-o uma linguagem muito flexível e fácil de aprender. Além disso, o JavaScript tem uma grande quantidade de **bibliotecas e frameworks, como o React, Angular e Vue**, que permitem que os desenvolvedores criem aplicativos complexos com facilidade. O JavaScript é uma das **linguagens de programação mais populares do mundo** e é amplamente utilizado em muitas áreas, incluindo desenvolvimento web, aplicativos de desktop e desenvolvimento móvel.



Diferença entre Java e Javascript



JAVA

- Fortemente Tipada;
- Linguagem de programação Orientada a Objetos;
- Necessita ser compilado;
- Cria aplicações independentes;
- Java executa suas aplicações em uma máquina virtual ou em browser;



JAVASCRIPT

- Fracamente Tipada;
- Linguagem de scripts Orientada a Objetos;
- É interpretado pelo navegador;
- Foi criada pela Netscape e não faz parte da plataforma Java;
- Não cria aplicações independentes;
- Se encontra dentro do documento HTML;

Linguagem PHP

- ▶ PHP é uma linguagem de programação de script de servidor de código aberto que foi criada em 1994 por Rasmus Lerdorf. Originalmente, o PHP significava "Personal Home Page", mas agora é conhecido como "Hypertext Preprocessor". A linguagem PHP é frequentemente usada para desenvolver aplicativos web dinâmicos e interativos que se comunicam com bancos de dados e outras tecnologias de back-end. Ela pode ser integrada facilmente com HTML e outras linguagens de marcação para produzir páginas web dinâmicas. **A linguagem PHP é executada em um servidor web, o que significa que o código PHP é executado no servidor antes de ser enviado para o navegador do usuário, tornando-a uma linguagem do lado do servidor.** O PHP é conhecido por sua ampla disponibilidade em vários sistemas operacionais e plataformas de servidor, tornando-o uma opção popular para hospedagem de sites. Além disso, o PHP tem uma grande quantidade de bibliotecas e frameworks, como o Laravel e o Zendframework, que ajudam os desenvolvedores a criar aplicativos web com facilidade. O PHP é uma das linguagens de programação mais populares para desenvolvimento web e é amplamente utilizado em muitos sites populares, como Facebook e Wikipedia.



Linguagem C#

- ▶ C# é uma linguagem de programação moderna e orientada a objetos desenvolvida pela Microsoft Corporation em 2000 como parte do framework .NET. **Ela é projetada para criar aplicativos em uma variedade de plataformas, incluindo desktop, web e dispositivos móveis.** O C# é uma linguagem de alto nível que oferece muitas ferramentas poderosas para o desenvolvedor, incluindo gerenciamento automático de memória, tipagem forte e recursos avançados de tratamento de exceções. O C# foi desenvolvido para ser fácil de aprender **e possui uma sintaxe semelhante a outras linguagens de programação populares, como Java e C++.** Ele também possui muitas bibliotecas e frameworks, como o .NET Framework e o ASP.NET, que ajudam os desenvolvedores a criar aplicativos com mais facilidade e rapidez. O C# é amplamente utilizado para desenvolver **aplicativos empresariais, jogos, aplicativos de desktop e aplicativos móveis.** Ele é uma das linguagens de programação mais populares atualmente e é frequentemente usado por desenvolvedores em todo o mundo.

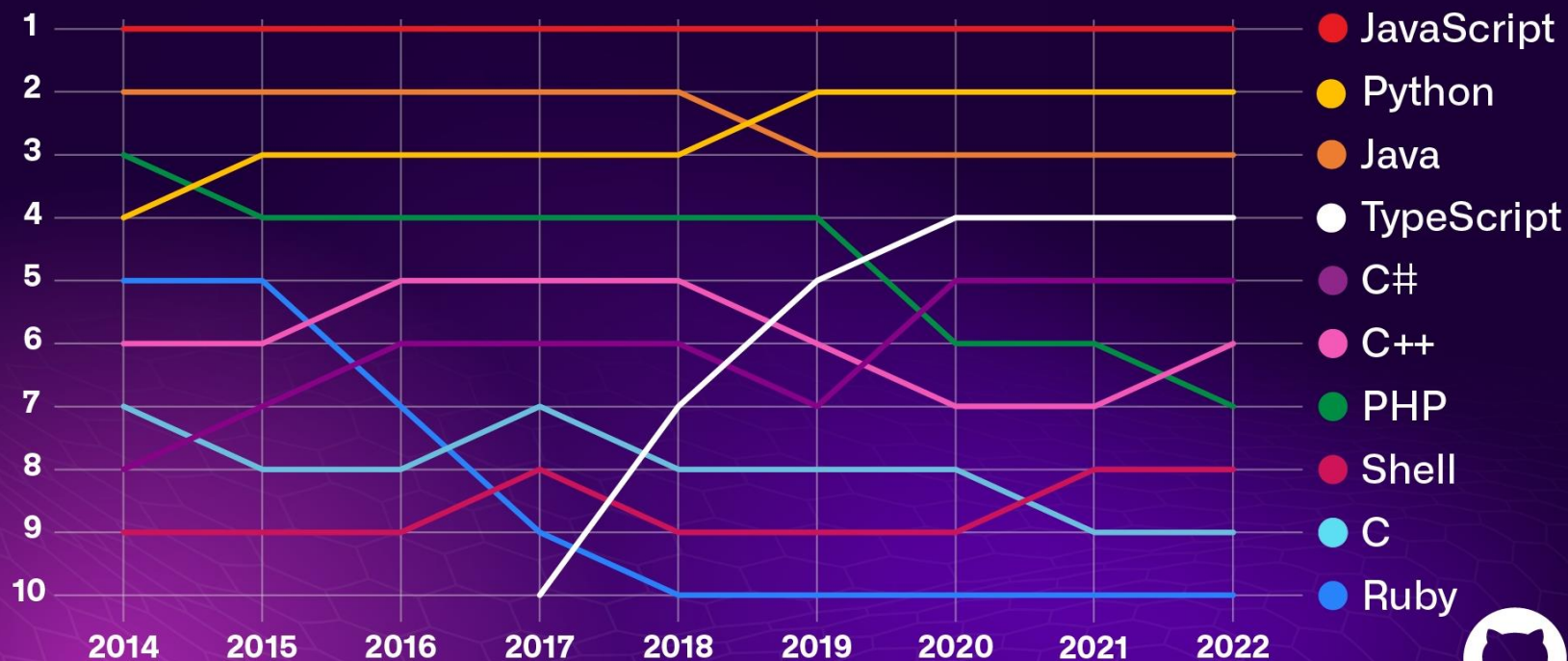


Linguagem Typescript

- ▶ TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft em 2012. É uma linguagem de script tipada que **estende a sintaxe do JavaScript adicionando tipos estáticos opcionais**. O TypeScript é projetado para aumentar a produtividade dos desenvolvedores e melhorar a manutenção do código em projetos grandes. Ele fornece uma compilação estática que pode pegar erros de sintaxe e semântica em tempo de compilação, o que ajuda a evitar problemas comuns que podem ocorrer em projetos JavaScript maiores e mais complexos. O TypeScript é compatível com todas as bibliotecas e frameworks JavaScript existentes e pode ser executado em qualquer navegador moderno, servidor ou plataforma que suporte JavaScript. Ele também fornece recursos adicionais para suporte a classes, interfaces, enums e outras estruturas de dados avançadas. **O TypeScript é uma escolha popular para o desenvolvimento de aplicativos web devido às suas melhorias de produtividade, tipagem forte e recursos avançados de codificação.**



Top 10 linguagens de programação 2022

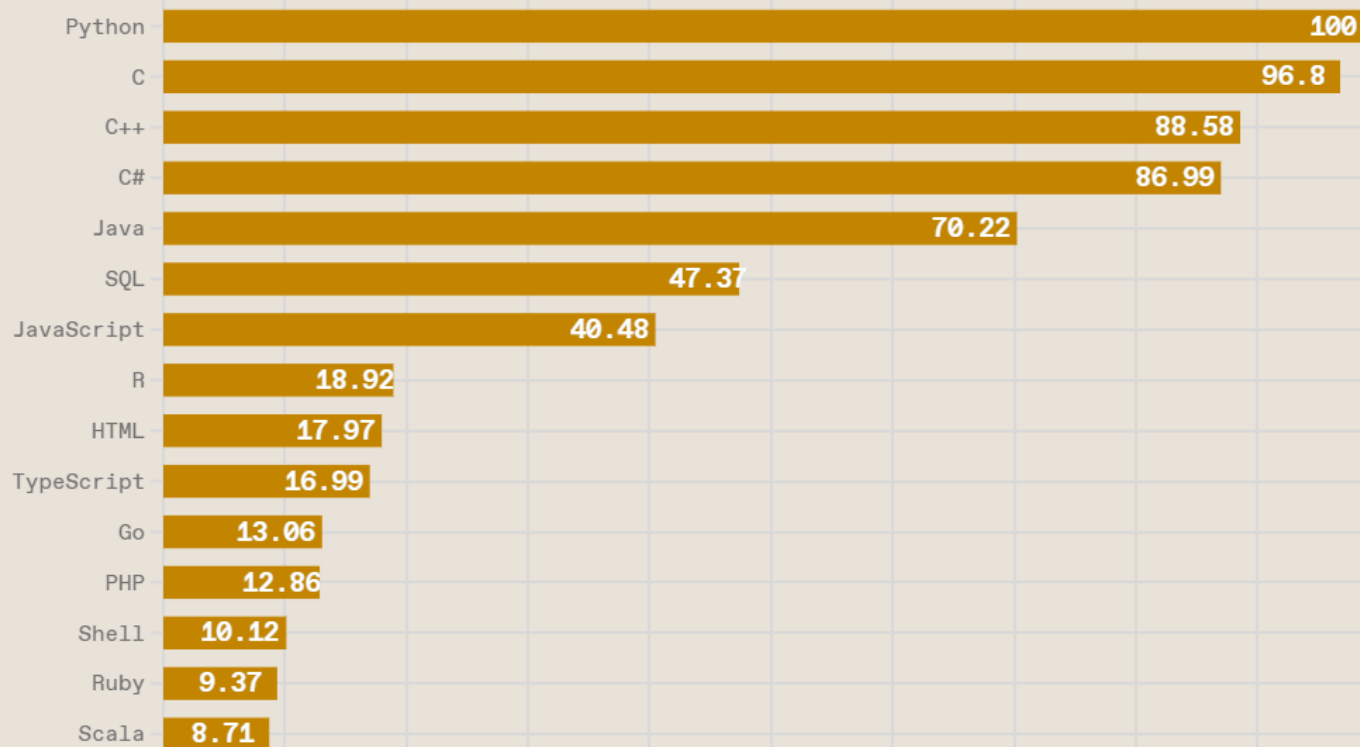


Linguagens mais utilizadas

Top Programming Languages 2022

Click a button to see a differently weighted ranking

Spectrum Jobs Trending



<https://spectrum.ieee.org/top-programming-languages-2022>

Compilação e interpretação

- ▶ Compilador é um software que converte código fonte escrito em uma linguagem de programação para código objeto, que é executado pelo computador. Ele verifica se o código está sintaticamente correto, traduz para uma linguagem intermediária e, em seguida, gera o código objeto final. O objetivo principal do compilador é transformar o código fonte em um formato executável que possa ser interpretado pelo computador.



Estruturas de dados

- ▶ Estrutura de dados é uma forma organizada de armazenar e gerenciar dados em um computador, de modo que possam ser acessados e manipulados com eficiência. **Essas estruturas incluem arrays, listas encadeadas, pilhas, filas, árvores e grafos, entre outras.** O objetivo da estrutura de dados é fornecer um meio de armazenar dados de maneira organizada para que possam ser processados eficientemente por algoritmos e programas de computador. A escolha da estrutura de dados adequada pode fazer uma grande diferença no desempenho e na eficiência de um programa.

Programação orientada a objetos

- ▶ Programação Orientada a Objetos (POO) é um paradigma de programação que se baseia na ideia de objetos, que são entidades que contêm dados e métodos associados a esses dados. Esses objetos interagem entre si para realizar tarefas e executar operações. **O POO é baseado em quatro princípios fundamentais: encapsulamento, herança, polimorfismo e abstração.** O encapsulamento envolve a proteção dos dados de um objeto, a herança permite que uma classe herde os atributos e métodos de outra classe, o polimorfismo permite que objetos de diferentes classes sejam tratados de forma semelhante e a abstração envolve a criação de classes e objetos que representam conceitos abstratos. .

Testes e depuração

- ▶ Testes para software são atividades realizadas para avaliar a qualidade e o desempenho de um programa de computador. Os testes podem ser realizados em diferentes níveis, como unitários, de integração, de sistema e de aceitação, e são usados para verificar se o software atende aos requisitos do usuário, se funciona corretamente em diferentes ambientes e se é seguro e confiável. **Os testes para software podem ser automatizados ou manuais, e utilizam técnicas como testes de caixa preta, testes de caixa branca, testes de estresse, testes de regressão e testes de usabilidade.** A realização de testes adequados para software é fundamental para garantir que o programa funcione corretamente e atenda às expectativas do usuário.

Algumas dicas

Ao se deparar com problema novo, tente entendê-lo

- ▶ Para auxiliar pense no seguinte:
 - O que deve descobrir ou calcular?
 - Quais são os dados disponíveis? São suficientes?
 - Quais são as condições necessárias e suficientes para resolver o problema?
 - Quais são as condições necessárias e suficientes para resolver o problema?
 - Faça um esboço informal de como ligar os dados com as condições.
 - Se possível, modele o problema de forma matemática.

Crie um plano de solução

- ▶ Consulte sua memória e verifique se já resolveu um problema similar. (Lições aprendidas, analogia)
- ▶ Verifique se é necessário introduzir algum elemento novo no problema, como um problema auxiliar;
- ▶ Se o problema for muito complicado, tente quebra-lo em partes menores e solucionar essas partes;
- ▶ É possível enxergar o problema de outra forma, de modo que seu entendimento se torne mais simples?

Formalize a solução

- ▶ Crie um algoritmo informal com passos que resolvam o problema.
- ▶ Verifique se cada passo desse algoritmo está correto;
- ▶ Escreva um algoritmo formalizado por meio de um fluxograma ou outra técnica de representação;

Exame dos resultados

- ▶ Teste o algoritmo com diversos dados de entrada e verifique os resultados (teste de mesa)
- ▶ Se o algoritmo não gerou resultado algum, o problema geralmente está na sua sintaxe ou nos comandos utilizados. Volte e tente encontrar o erro.
- ▶ Se o algoritmo gerou resultados, estes estão corretos? Analise suas consistência;
- ▶ Se não estão corretos, alguma condição, operação ou ordem das operações podem estar incorretas. Volte e tente encontrar o erro

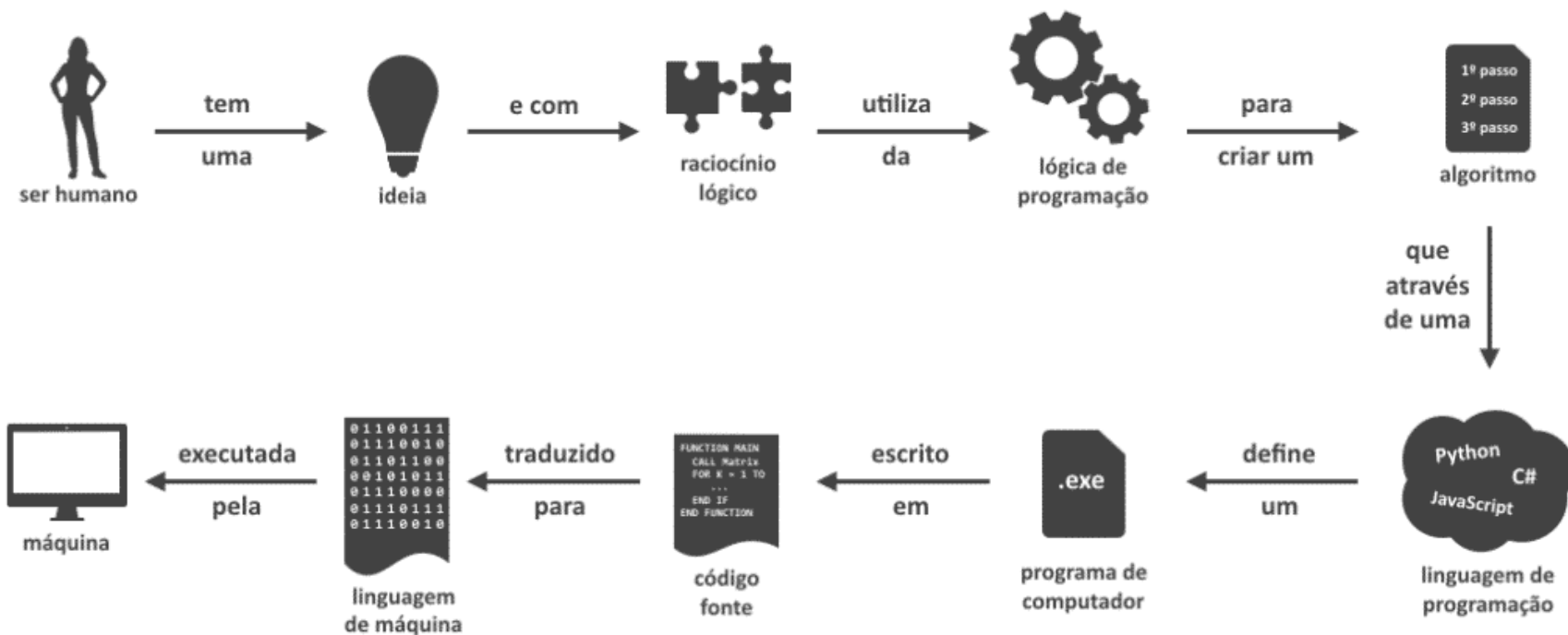
Otimização da solução

- ▶ É possível melhorar o algoritmo?
- ▶ É possível reduzir o número de passos?
- ▶ É possível conseguir uma solução ótima?

Sempre se atente a sintaxe e a semântica

- ▶ A **sintaxe** é a **gramática** da linguagem de programação e define como o código deve ser escrito, incluindo o uso correto de palavras-chave, operadores, parênteses, colchetes, chaves e outros símbolos. Para definir a sintaxe de um código, é necessário conhecer a gramática da linguagem de programação em questão e seguir as regras estabelecidas por ela;
- ▶ A **Semântica** define o **comportamento** que o programa deve ter em relação aos dados de entrada e saída, bem como as operações e algoritmos que são realizados sobre esses dados. A semântica está relacionada com a lógica e o raciocínio por trás do código, e envolve a compreensão dos conceitos e regras da linguagem de programação utilizada

Fluxo de trabalho



Resumo da aula

- ▶ Entendemos os conceitos de linguagem de programação e seu pensamento lógico;
- ▶ Entendemos os principais aspectos necessário para o desenvolvimento de software.
- ▶ Conhecemos as principais linguagens de software.
- ▶ Para próxima aula vamos trabalhar com arquitetura de banco de dados

Obrigado!!!

Profº Jânio Eduardo

janio.vasconcellos@gmail.com